

Amendments to the Claims

D2
E1
Claims 1 - 24 (canceled)

1 Claim 25 (currently amended): A method of searching and updating indexes to a database data
2 store in a multi-processing environment, comprising the steps of:
3 maintaining two databases indexes to the data store, database1 a first index for searching
4 and a database2 second index for updating, updating:
5 responsive to each update of after updating database2 the second index, switching the
6 databases indexes so that the first index becomes the second index and the updated database
7 second index becomes database1 the first index; and the non-updated database becomes
8 database2;
9 allowing searches that are in progress at the time of using the first index, before the
10 switching, to of the database to continue until completion after the switching, using what is then
11 the second index; in now database2;
12 allowing after the switching, initiating new searches using what is then the first index; to
13 initiate in now database1;
14 when all searches in database2 what is then, after the switching, the second index have
15 completed, updating database2 what is then the second index in an identical manner as the update
16 to which the switching step was responsive; and with the prior update that caused to the last
17 database switch;
18 preventing another operation of the switching step database switch until completion of the
19 step of updating the second index in the identical manner, after the last step has completed.

Serial No. 09/753,992

-3-

Docket RSW919990130US1

E1
DJ
Claim 26 (canceled)

1 Claim 27 (currently amended): A program product storage medium containing computer
2 instructions that when executed in a computer perform a method of searching and updating
3 indexes to a database data store in a multi-processing environment, the method comprising the
4 steps of:

5 maintaining two databases indexes to the data store, database1 a first index for searching
6 and a database2 second index for updating, updating:

7 responsive to each update of after updating database2 the second index, switching the
8 databases indexes so that the first index becomes the second index and the updated database
9 second index becomes database1 the first index; and the non-updated database becomes
10 database2;

11 allowing searches that are in progress at the time of using the first index, before the
12 switching to of the database to continue until completion after the switching, using what is then
13 the second index; in now database2;

14 after the switching, initiating allowing new searches using what is then the first index; to
15 initiate in now database1;

16 when all searches in database2 what is then, after the switching, the second index have
17 completed, updating database2 what is then the second index in an identical manner as the update
18 to which the switching step was responsive; and with the prior update that caused to the last
19 database switch;

Serial No. 09/753,992

-4-

Docket RSW919990130US1

E1 20 preventing another operation of the switching step database switch until after the last step
21 has completed: completion of the step of updating the second index in the identical manner.

DJ Claim 28 (canceled)

1 Claim 29 (currently amended): A computer program product for serializing data structure
2 retrievals and updates in a multi-processing computer system, the computer program product
3 embodied on one or more computer-readable media and comprising:
4 computer-readable program code means for creating two identical database data
5 structures, each representing an initial state for accessing a single copy of stored data;
6 computer-readable program code means for performing searches against a first of the two
7 databases data structures, said the computer-readable program code means for performing
8 searches further comprising a first program instruction for incrementing a search use count for the
9 first data structure atomically during each search to insure ensure no interference from other
10 processes during that search and a second instruction for decrementing the search use count for
11 the first data structure atomically after performing the each search;
12 computer-readable program code means for performing a first update against a second of
13 the two databases data structures, yielding a revised database data structure;
14 computer-readable program code means for switching the first database data structure and
15 the revised database data structure, responsive to completion of the computer-readable program
16 code means for performing the first update, such that the first database data structure becomes
17 the second database data structure and the revised database data structure becomes the first

Serial No. 09/753,992

-5-

Docket RSW919990130US1

E1

18 database data structure, said the computer-readable program code means for switching the
19 databases data structures further comprising a third instruction for re-ordering database data
20 structure pointers atomically to prevent interference from other processes during operation of the
21 computer-readable program code means for switching; and

22 computer-readable program code means for performing applying, after operation of the
23 computer-readable program code means for switching, a second the first update against the
24 second database data structure, yielding a synchronized database second data structure that is
25 structurally identical to the first database data structure;

26 said the computer-readable program code means for performing searches further
27 comprising computer-readable program code means for activating the last-mentioned computer-
28 readable program code means for performing a second applying the first update against the
29 second database data structure when the search use count for the now second database data
30 structure has a value indicating that no searches are being performed against the second data
31 structure is zero.

1 Claim 30 (currently amended): The computer program product according to Claim 29, further
2 comprising:

3 computer-readable program code means for obtaining an exclusive lock on the second
4 data structure prior to operation of the computer-readable program code means for performing
5 the first update; and

6 computer-readable program code means for releasing the exclusive lock after operation of
7 the computer-readable program code means for performing the second applying the first update

E1
8 and the computer-readable program code means for switching.

DZ
1 Claim 31 (currently amended): The computer program product according to Claim 29, wherein
2 the computer-readable program code means for performing the first update further comprises
3 computer-readable program code means for queuing a transaction that specifies one or more data
4 structure traversals and one or more data structure modifications that were performed to yield the
5 revised data structure, and wherein the computer-readable program code means for performing
6 the second applying the first update further comprises computer-readable program code means
7 for performing the one or more data structure traversals and the one or more modifications
8 specified in the queued transaction applying the queued transaction against the second database
9 data structure that results from operation of the computer-readable program code means for
10 switching.

1 Claim 32 (currently amended): The computer program product according to Claim 29, further
2 comprising computer-readable program code means for performing a subsequent update against
3 the synchronized database second data structure that results from operation of the computer-
4 readable program code means for performing the second applying the first update; and wherein
5 operation of the computer-readable program code means for performing the subsequent update
6 causes another operation of the computer-readable program code means for switching and the
7 computer-readable program code means for applying.

1 Claim 33 (currently amended): A computer system for serializing data structure retrievals and
Serial No. 09/753,992

E/
D/J
2 updates in a multi-processing computer system, the computer system comprising:

3 means for creating two identical database data structures, each representing an initial state

4 for accessing a single copy of stored data;

5 means for performing searches against a first of the two databases data structures, said the

6 means for performing searches further comprising means for incrementing a search use count for

7 the first data structure each search atomically to insure during each search to ensure no

8 interference from other processes during that search and means for atomically decrementing the

9 search use count for the first data structure after performing the search, each search:

10 means for performing a first update against a second of the two databases data structures,

11 yielding a revised database data structure;

12 means for switching the first database data structure and the revised database data

13 structure, responsive to completion of the means for performing the first update, such that the

14 first database data structure becomes the second database data structure and the revised database

15 data structure becomes the first database data structure, said the means for switching the

16 databases data structures further comprising means for re-ordering database data structure

17 pointers atomically to prevent interference from other processes during operation of the means for

18 switching; and

19 means for applying performing, after switching the databases data structures, a second the

20 first update against the second database data structure, yielding a synchronized database second

21 data structure that is structurally identical to the first database data structure;

22 said the means for performing searches further comprising means for activating the last-

23 mentioned means for performing a second applying the first update against the second database

E/

24 data structure when the search use count for the now second database is zero database data
25 structure has a value indicating that no searches are being performed against the second data
26 structure.

DJ

1 Claim 34 (currently amended): The system according to Claim 33, further comprising:
2 means for obtaining an exclusive lock on the second data structure prior to operation of
3 the means for performing the first update; and
4 means for releasing the exclusive lock after operation of the means for applying the first
5 performing the second update and the means for switching.

1 Claim 35 (currently amended): The system according to Claim 33, wherein the means for
2 performing the first update further comprises means for queuing a transaction that specifies one
3 or more data structure traversals and one or more data structure modifications that were
4 performed to yield the revised data structure, and wherein the means for performing the second
5 applying the first update further comprises means for performing the one or more data structure
6 traversals and the one or more data structure modifications specified in the queued transaction
7 applying the queued transaction against the second database data structure that results from
8 operation of the means for switching.

1 Claim 36 (currently amended): The system according to Claim 33, further comprising means for
2 performing a subsequent update against the synchronized database second data structure that
3 results from operation of the means for performing the second applying the first update; and

Serial No. 09/753,992

-9-

Docket RSW919990130US1

EI
DZ 5
4 wherein operation of the means for performing the subsequent update causes another operation of
the means for switching and the means for applying.

1 Claim 37 (currently amended): A method for serializing data structure retrievals and updates in a
2 multi-processing computer system, comprising the steps of:

3 creating two identical database data structures, each representing an initial state for
4 accessing a single copy of stored data;
5 performing searches against a first of the two databases data structures, the said
6 performing searches step further comprising the step of incrementing a search use count for the
7 first data structure atomically during each search to ensure while insuring no interference from
8 other processes during the search and the step of decrementing the search use count for the first
9 data structure atomically after performing the each search also while insuring no interference from
10 other processes;

11 performing a first update against a second of the two databases data structures, yielding a
12 revised database data structure;

13 switching the first database data structure and the revised database data structure,
14 responsive to completion of the step of performing the first update, such that the first database
15 data structure becomes the second database data structure and the revised database data structure
16 becomes the first database data structure, said the step of switching the databases data structures
17 further comprising the step of re-ordering database data structure pointers while preventing
18 atomically to prevent interference from other processes during operation of the switching step;
19 and

E1
20 performing applying, after the switching step of databases, a second the first update
21 against the second database data structure, yielding a synchronized database second data structure
22 that is structurally identical to the first database data structure;
23 said the step of performing searches further comprising the step of activating the last-
24 mentioned step of applying the first update for performing a second update against the second
25 database data structure when the search use count for the now second database is zero data
26 structure has a value indicating that no searches are being performed against the second data
27 structure.

1 Claim 38 (currently amended): The method according to Claim 37, further comprising steps of:

2 obtaining an exclusive lock on the second data structure prior to performing the first
3 update; and

4 releasing the exclusive lock after performing the second applying the first update and the
5 switching.

1 Claim 39 (currently amended): The method according to Claim 37, wherein the step of
2 performing the first update further comprises the step of queuing a transaction that specifies one
3 or more data structure traversals and one or more data structure modifications that were
4 performed to yield the revised data structure, and wherein the step of performing the second
5 applying the first update further comprises the step of performing the one or more data structure
6 traversals and the one or more data structure modifications specified in applying the queued
7 transaction against the second database data structure that results from operation of the switching

8 step.

D2
1 Claim 40 (currently amended): The method according to Claim 37, further comprising the step of
2 performing a subsequent update against the ~~synchronized database~~ second data structure that
3 results from performing the second applying the first update, and wherein the step of performing
4 the subsequent update causes repeating the switching step and the applying step.

D3
1 Claim 41 (new): A method for serializing data retrievals and updates in a computing
2 environment, comprising steps of:

3 creating two identical indexes, each representing an initial state for accessing stored data
4 and each indexing a single copy of the stored data;

5 performing searches against a first of the two indexes;

6 performing a first update against a second of the two indexes, yielding a revised index;

7 serializing information on how the first update affected the second index, including how
8 the second index was traversed for making the first update and how the second index was
9 modified in the first update;

10 switching the first index and the revised index, responsive to performing the first update,
11 such that the first index becomes the second index and the revised index becomes the first index;

12 applying, after the switching step, the serialized information to the second index, using the
13 information about how the second index was traversed and modified to efficiently traverse and
14 modify the newly-switched second index, thereby yielding a second index that is synchronized
15 with, and structurally identical to, the first index; and

Serial No. 09/753,992

-12-

Docket RSW919990130US1

16 performing subsequent searches against the first index.

D31
1 Claim 42 (new): The method according to Claim 41, further comprising the step of performing a
2 subsequent update against the second index that results from applying the serialized information;
3 and wherein the step of performing the subsequent update causes repeating the serializing,
4 switching, and applying steps.

1 Claim 43 (new): A method of serializing access to data in a computing system, comprising steps
2 of:

3 maintaining two trees as indexes to the data, a first of which is used for searches and a
4 second of which is used for update operations, each tree having a use count associated therewith;

5 carrying out searches using the search tree, further comprising the steps of:

6 determining, for each new search request, which of the trees is currently the search
7 tree;

8 incrementing the use count for the search tree;

9 performing the new search request using the search tree; and

10 decrementing the use count for the search tree, responsive to completion of the
11 performing step; and

12 carrying out each update using the update tree, further comprising the steps of:

13 determining which of the trees is currently the update tree;

14 performing an update against the update tree;

15 serializing a record of how the update affected the update tree;

EI
16 switching the update tree to become the search tree and the search tree to become
17 the update tree, responsive to completion of the steps of performing the update and serializing the
18 record; and

DJ
19 applying the serialized record to the newly-switched update tree, provided that the
20 use count for the newly-switched update tree has reached a value that indicates that no search
21 requests are currently being performed against this newly-switched update tree, delaying the step
22 of applying the serialized record if necessary until the use count for the newly-switched update
23 tree has reached this value, and wherein the step of applying the serialized record ensures that
24 both the search tree and the update tree reflect each update.

1 Claim 44 (new): The method according to Claim 41, wherein the indexes are implemented as
2 trees.

1 Claim 45 (new): The method according to Claim 41, wherein the indexes are implemented as
2 hash tables.